PAT-NO: JP02000357097A
DOCUMENT-IDENTIFIER: JP 2000357097 A
TITLE: PROGRAM MANAGING METHOD
PUBN-DATE: December 26, 2000

INVENTOR-INFORMATION:

NAME	COUNTRY
AOKI, TOSHIYUKI	N/A

ASSIGNEE-INFORMATION:

NAME	COUNTRY
NEC CORP	N/A

APPL-NO: JP11169426
APPL-DATE: June 16, 1999

INT-CL (IPC): G06F009/45 , G06F009/06

ABSTRACT:

PROBLEM TO BE SOLVED: To specify a relation among a source program, an object program and a run unit program.

SOLUTION: According to an object program name, a compiler name, a compile option, a compiler version and compile date/time information as compile information 23 extracted from an object program 21 of an investigation object, a source program 11 is compiled, an object program 12 of the output result provided by compiling is compared with the object program 21 of the investigation object and it is discriminated whether the object program 21 of the investigation object is compiled from the source program 11 or not.

(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（A）　(11)特許出願公開番号
特開2000－357097
（P2000－357097A）
(43)公開日　平成12年12月26日(2000.12.26)

(51)Int.Cl.'		識別記号	FI		テーマコード'(参考)
G06F	9/45		G06F	9/44	322J　5B076
	9/06	410		9/06	410P　5B081

審査請求　有　　請求項の数6　OL　（全 6 頁）

(21)出願番号　特願平11－169426

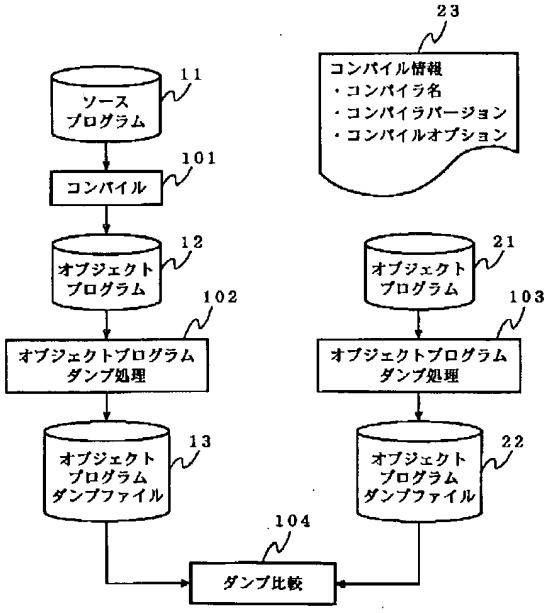
(22)出願日　平成11年6月16日(1999.6.16)

(71)出願人　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(72)発明者　青木　俊之
東京都港区芝五丁目７番１号　日本電気株式会社内
(74)代理人　100086645
弁理士　岩佐　義幸

Ｆターム(参考)　5B076 AB15 AC08
5B081 BB08 CC41

(54)【発明の名称】　プログラム管理方法

(57)【要約】
【課題】　ソースプログラム、オブジェクトプログラム、ランユニットプログラム間の関連付けを特定する。
【解決手段】　調査対象のオブジェクトプログラム２１から抽出したコンパイル情報２３であるオブジェクトプログラム名、コンパイラ名、コンパイルオプション、コンパイラバージョン、コンパイル日時情報に従ってソースプログラム１１をコンパイルし、コンパイルして得られた出力結果のオブジェクトプログラム１２と、調査対象のオブジェクトプログラム２１とを比較して、調査対象のオブジェクトプログラム２１がソースプログラム１１からコンパイルされたものかどうかを判定する。



1

【特許請求の範囲】
【請求項1】ソースプログラムとオブジェクトプログラムとの対応を管理するプログラム管理方法において、
調査対象のオブジェクトプログラムから抽出したコンパイル情報に従ってソースプログラムをコンパイルし、コンパイルして得られた出力結果のオブジェクトプログラムと、前記調査対象のオブジェクトプログラムとを比較して、調査対象のオブジェクトプログラムが前記ソースプログラムからコンパイルされたものかどうかを判定することを特徴とするプログラム管理方法。
【請求項2】前記コンパイル情報は、オブジェクトプログラム名、コンパイラ名、コンパイルオプション、コンパイラバージョン、コンパイル日時であることを特徴とする請求項1に記載のプログラム管理方法。
【請求項3】前記調査対象のオブジェクトプログラムと前記ソースプログラムをコンパイルして出力されたオブジェクトプログラムをそれぞれダンプファイルに編集出力し、ダンプファイル同士を比較することを特徴とする請求項1または2に記載のプログラム管理方法。
【請求項4】前記ダンプファイル同士をシーケンシャルにレコード単位に比較することを特徴とする請求項3に記載のプログラム管理方法。
【請求項5】オブジェクトプログラムとランユニットプログラムとの対応を管理するプログラム管理方法において、
ランユニットプログラムを読み込んで、ランユニットプログラム中に含まれるオブジェクト情報を抽出し、調査対象となるオブジェクトプログラムのオブジェクト情報と比較し、前記ランユニットプログラムが前記調査対象となるオブジェクトプログラムからリンクされたものかどうかを判定することを特徴とするプログラム管理方法。
【請求項6】前記オブジェクト情報は、オブジェクトプログラム名、コンパイル日時であることを特徴とする請求項5に記載のプログラム管理方法。
【発明の詳細な説明】
【0001】
【発明の属する技術分野】この発明は、ソースプログラム、オブジェクトプログラム、ランユニットプログラム間を関連付けるプログラム管理方法に関する。
【0002】
【従来の技術】ソースプログラムからコンパイラを通して、機械語に変換した結果がオブジェクトプログラムである。ソースプログラムとそのオブジェクトプログラムは、それぞれが独立のファイルに管理されているので、その対応付けを管理するために、コンパイル時に出力された実行レポートや翻訳リストを保管して、そのレポート上に存在するソースプログラム名およびソースプログラムファイル名、オブジェクトファイル名、オブジェクト名、コンパイル日時、バージョン情報などを手掛かりに対応付けを見極めている（リスト管理）。

2

【0003】また、上記方法でソースプログラムおよびオブジェクトプログラムを管理する以外に、従来の技術として、コンパイルやリンクをした際、自動的に管理情報がディクショナリファイルに登録され、そのディクショナリ情報を見ることによってプログラムを管理する方法もある（ディクショナリ管理）。
【0004】
【発明が解決しようとする課題】しかし、従来のこれら管理方法には、次のような問題点がある。
【0005】まず、（リスト管理）方法には、コンパイルリストやリンクマップの紛失時に、オブジェクトプログラムの元ソースプログラムがどれか、また、ランユニットプログラムの元オブジェクトプログラムがどれか、それを調査して対応付けを行う手段が無いという問題点がある。
【0006】これは、第1には、ソースプログラム本体には、出力したオブジェクトプログラムに関するファイル名、コンパイル日時などの管理情報がないことによる。たとえコメントとして記載されていても、そのコメント自体が改版されていなかったり、不十分であったりして信用できないことがあり、また、オブジェクトプログラム本体には、その入力となったソースプログラムとの関連付けができるような、例えば、ソースファイル名、ソースプログラムテキストなどの情報が無いからである。
【0007】第2には、コンパイルリストを紛失した場合以外にも、ソースプログラムとオブジェクトプログラムとの関連付けは、下記の場合に失われることである。
【0008】最初にオブジェクトプログラムが出力されたファイルから、オブジェクトが別のオブジェクトファイルに移動し使用される場合がある。これは利用者がテストなどの際利用しやすい環境に移動するためであるが、その移動は、オブジェクトプログラムライブラリアンで行われ、その際に元のソースプログラムやオブジェクトプログラム、ランユニットプログラム管理体系との関連付けを情報として残すことを利用者が忘れ、運用者管理基準外の方法で本番システムへのプログラムリリースする場合などがあることである。特に緊急トラブル対応時などこのような手段で本番システムの変更を行うこともある。このような管理規則を外れたプログラムが、20～30年という長い年月を経た膨大な既存プログラム資産の内には非常に多く存在していることが多い。
【0009】第3には、オブジェクトプログラムは、いろいろなランユニットファイルにリンクされ使用される場合があるので、ランユニットファイル名自体をオブジェクトプログラム上には持っていないし、従来ランユニットプログラムがどのオブジェクトプログラムから生成されたものかを知る手段が無かったということである。
【0010】次に、（ディクショナリ管理）方法では、

（ディクショナリ管理）を導入する以前から既に存在している既存プログラム資産について、ディクショナリ管理を適用し難いという問題点がある。

【００１１】新規開発のプログラムについては、新規にプログラム管理体系を定め、（ディクショナリ管理）の環境を作成し、その環境上でプログラム開発すれば良いので比較的容易に（ディクショナリ管理）への移行が可能であるが、既存プログラムについては、既存のプログラム開発環境を（ディクショナリ管理）できる開発環境に変更することが必要になる。

【００１２】プログラム開発環境の変更時には、個別システム内で閉じていたプログラムを他システムと重複・相反しないように新規に標準的ルールを定め、修正・変更・新規登録することが必要になる。

【００１３】この場合、ソースプログラムとオブジェクトプログラムの管理情報が紛失している場合やプログラム資産の名称の重複、また、同一プログラムを個別に改造して使用している場合もあるので、それらを一意に整理することになる。そのためには、ライブラリファイルを全サーチし、同一プログラムと考えられるものを全て洗い出し、重複して存在しているソースプログラムやオブジェクトファイルを本番システムで使用しているものとテスト時に使用したもの、緊急のトラブル対応に一時的に修正を加えたもの、開発段階の消し忘れなど、１つ１つ仕様書やプログラムのコメントを頼りに見直さなければならない。また、新たに仕様書を興し、プログラム登録管理をやり直す場合もある。

【００１４】プログラム名称に変更が発生する場合には、ただ名称の変更に留まらず、本番システム上で動作しているランユニットプログラムの名称も変更するので、具体的には（ディクショナリ管理）環境化で再コンパイル～リンクして、ランユニットプログラムを作成後、評価して、本番システム上でのランユニットプログラム変更作業を行うという一連の多大な工数が必要である。

【００１５】（ディクショナリ管理）は、（ディクショナリ管理）導入以降、その仕組みを組み込めたソースプログラム、オブジェクトプログラム、ランユニットプログラム間のみでの関連性が失われないように、プログラム管理データの紛失を防止することを目的としているシステムであって、既に、オブジェクトプログラム、ランユニットプログラム間の関連性が不明となった状態には無効である。

【００１６】この発明の目的は、既に、ソースプログラム、オブジェクトプログラム、ランユニットプログラム間の関連性が失われていた場合に、それらの間の関連付けを特定できるようにして、プログラム管理情報を完全化するためのプログラム管理方法を提供することである。

【００１７】

【課題を解決するための手段】この発明は、ソースプログラムとオブジェクトプログラムとの対応を管理するプログラム管理方法において、調査対象のオブジェクトプログラムから抽出したコンパイル情報に従ってソースプログラムをコンパイルして、その出力結果のオブジェクトプログラムと、前記調査対象のオブジェクトプログラムとを比較して、調査対象のオブジェクトプログラムが前記ソースプログラムからコンパイルされたものかどうかを判定することを特徴とする。

【００１８】また、この発明は、オブジェクトプログラムとランユニットプログラムとの対応を管理するプログラム管理方法において、ランユニットプログラムを読み込んで、ランユニットプログラム中に含まれるオブジェクト情報を抽出し、調査対象となるオブジェクトプログラムのオブジェクト情報と比較し、前記ランユニットプログラムが前記調査対象となるオブジェクトプログラムからリンクされたものかどうかを判定することを特徴とする。

【００１９】この発明は、ソースプログラム、オブジェクトプログラム、ランユニットプログラム本体のみの情報から、それらが同一ソースプログラムからの生成物として関連付けられるかどうかを一意的に判定することができる。

【００２０】

【発明の実施の形態】次に、この発明のプログラム管理方法の実施の形態について図面を参照して説明する。

【００２１】図１は、この発明の実施の形態であり、ソースプログラムとオブジェクトプログラムとの関連付けを特定する処理手順を説明する図である。

【００２２】上記プログラムを関連付けを特定するための判定方法の概要としては、ソースプログラム１１をコンパイルした結果のオブジェクトプログラム１２と、オブジェクトプログラム２１とを比較し、それらが同一ソースプログラム１１の生成物であることを証明する。

【００２３】コンパイラの種類、コンパイルオプション、コンパイラバージョンによって、同一ソースプログラムからでも生成されるオブジェクトは異なり、単に比較できないので、これらコンパイル情報３０をオブジェクトプログラム２１からコンパイル前に抽出しておき、その情報に従ってコンパイル実施後に出力されたオブジェクトプログラム１２と比較する。

【００２４】これらコンパイル情報３０は、コンパイラ提供メーカーがコンパイラに関してのエラーが発生した場合に対応するためや、オブジェクトをライブラリ管理する必要上、一般的にオブジェクト中に情報として保持されている。

【００２５】次に、オブジェクトプログラム２１と、コンパイルしてできたオブジェクトプログラム１２とを各々ダンプ編集出力し、その編集データをレコード単位に比較し、相違のあるレコードを出力する。比較の結果、

コンパイル日時だけが異なれば、オブジェクトは一致したと見なせ、これにより、オブジェクトプログラム２１は、ソースプログラム１１をコンパイルしたオブジェクトであると判定できる。

【００２６】次に、この発明の実施の形態の処理動作を図１および図２を参照して詳細に説明する。図２は、オブジェクトプログラムからコンパイル情報を抽出する処理手順を説明する図である。

【００２７】まず、オブジェクトプログラムからコンパイル情報を抽出する手順を説明する。図２において、ステップ２０１～２０７の処理手順を介して、オブジェクトプログラム２１を入力として、オブジェクトプログラム中のオブジェクトプログラム名、コンパイラ名、コンパイラバージョン、コンパイル日時情報を抽出し、また、コンパイルオプション情報については、メモリ中のコンパイルオプションテーブル４２とオブジェクトプログラム中のコンパイルオプションに相当するデータを突き合わせて、利用者が見て指定できるコンパイラに対するコンパイルオプションとして、コンパイル情報２３をコンパイル情報抽出ファイル２５に出力する。

【００２８】図２で示した処理手順の目的は、コンパイラの種類、コンパイラオプション、コンパイラバージョンの相違によって出力されるオブジェクトが命令もオブジェクトサイズも異なるため、これらコンパイル情報を明確にすることである。

【００２９】オブジェクトプログラム上のこれらコンパイル情報を明確にした上で、ソースプログラムからオブジェクトプログラムを生成し、他オブジェクトプログラムとの比較に用いることが必要である。

【００３０】次に、図１のステップ１０１、ステップ１０２の処理手順について説明する。

【００３１】ソースプログラム１１を入力として、図２の処理手順で得られたコンパイル情報２３のうち、コンパイラ名、コンパイラバージョン、コンパイルオプションに従って、該当するコンパイル（ステップ１０１）を実行し、オブジェクトプログラム１２を出力する。それをオブジェクトプログラムダンプ処理（ステップ１０２）でオブジェクトプログラムダンプファイル１３に編集出力する。

【００３２】オブジェクトプログラムダンプ処理では、オブジェクトプログラムをデータ部とコード部に分けて、使用セグメント名情報を編集付加してシーケンシャルファイルに出力する。

【００３３】オブジェクトプログラム同士を直接比較せず、オブジェクトプログラムダンプ処理でシーケンシャルファイルに編集出力したもの同士を比較する理由は、相違するデータがあった場合に、利用者が比較して見やすくするためである。

【００３４】次に、図１のステップ１０３の処理手順について説明する。

【００３５】オブジェクトプログラム２１に対してオブジェクトプログラムダンプ処理（ステップ１０３）を実行し、オブジェクトプログラムダンプファイル２２を作成する。

【００３６】このステップ１０１～１０３の処理手順でできた２つの、ダンプファイル１３とダンプファイル２２をステップ１０４でシーケンシャルにレコード単位で比較し、一致しないレコードについては、２つのオブジェクトプログラムダンプのレコードイメージを比較するため並べてリスト出力する。このステップ１０４の処理では、コンパイル日時は一致しないので、それ以外のデータが一致すれば、オブジェクトプログラム２１の元ソースプログラムがソースプログラム１１であることが判定できたことになる。

【００３７】次に、ランユニットプログラムがオブジェクトプログラムからの生成物であるかどうかを判定する処理手順を説明する。図３は、ランユニットプログラムとオブジェクトプログラムとの関連付けを特定する処理手順を説明する図である。

【００３８】上記プログラムを関連付けを特定するための判定方法の概要としては、ランユニットプログラム３１中には、オブジェクトプログラムのオブジェクトイメージがリンク時に取り込まれているので、そのオブジェクトに含まれるオブジェクト情報であるオブジェクトプログラム名とコンパイル日時を、調査対象のオブジェクトプログラム３２内のオブジェクト情報と比較し、一致すれば、ランユニットプログラム３１は、オブジェクトプログラム３２からリンクしてできた生成物であると判定する。

【００３９】図３のステップ３０１～３０４は、ランユニットプログラム３１のリンク前の元オブジェクトプログラムがオブジェクトプログラム３２であるかどうかを確認するための処理手順である。この処理手順の目的は、オブジェクトプログラムが複数あってどれがマスターであるか決められない場合、後でオブジェクトプログラムとソースプログラムの突き合わせを行うこともできなくなるので、本番システム上で稼働しているランユニットプログラムを基準にしてオブジェクトプログラムを特定することである。

【００４０】ステップ３０１により、ランユニットプログラム３１を入力として、ランユニットプログラム中に含まれるオブジェクトプログラム名、コンパイル日時を抽出する。

【００４１】次に、オブジェクトプログラム３２にコンパイル情報抽出処理を実施し（ステップ３０２）、オブジェクトプログラム中に含まれるオブジェクトプログラム名とコンパイル日時を抽出する。

【００４２】この２つのオブジェクトプログラム名、コンパイル日時を読み込み（ステップ３０３）、オブジェクトプログラム名、コンパイル日時が一致しているかど

うかをステップ３０４で比較し、一致すれば、ランユニットプログラム３１は、オブジェクトプログラム３２のリンク前の元オブジェクトプログラムであると判定できる。

【００４３】また、一致しなければ、ランユニットプログラム中のオブジェクトプログラム名、コンパイル日時、オブジェクトプログラムのオブジェクトプログラム名、コンパイル日時を出力し、利用者の参考情報とする。

【００４４】

【発明の効果】以上説明したように、この発明は、ソースプログラムとオブジェクトプログラムとランユニットプログラムの関連付けを、それらの管理情報が消えてしまっていても完全に保証することができる。これによって、プログラム開発環境を整理することができるので、誤ったソースプログラムやオブジェクトプログラムを使用して開発を進めたためのトラブルが発生することが無くなる。

【００４５】例えば、ランユニットプログラムは１つだけで、オブジェクトプログラムやソースプログラムが複数存在し、どれがランユニットプログラムに対する元オブジェクトプログラム、元ソースプログラムであるか分からない場合にこの方法を用いれば、ランユニットプログラムからオブジェクトプログラムが特定され、そのオブジェクトプログラムからソースプログラムが特定できる。

【００４６】２０００年問題など過去の既存資産を大規模に見直す状況が発生して、１０～３０年前のプログラム資産で、現在ではソースプログラム、オブジェクトプログラム、ランユニットプログラムの対応付けができなくなったプログラムが多々ある場合などに、その整理作業や開発作業の大幅な工数低減を図ることができる。

【００４７】また、管理情報のない既存資産の開発環境を整理できるため、これまでソースプログラム、オブジェクトプログラム、ランユニットプログラム間の関連付けが取れなかったため、（ディクショナリ管理）を適用できなかったプログラムに対して、（ディクショナリ管理）を適用できるようになるため、今後のプログラム資産管理を容易にすることが可能になる。

【図面の簡単な説明】

【図１】ソースプログラムとオブジェクトプログラムとの関連付けを特定する処理手順を説明する図である。

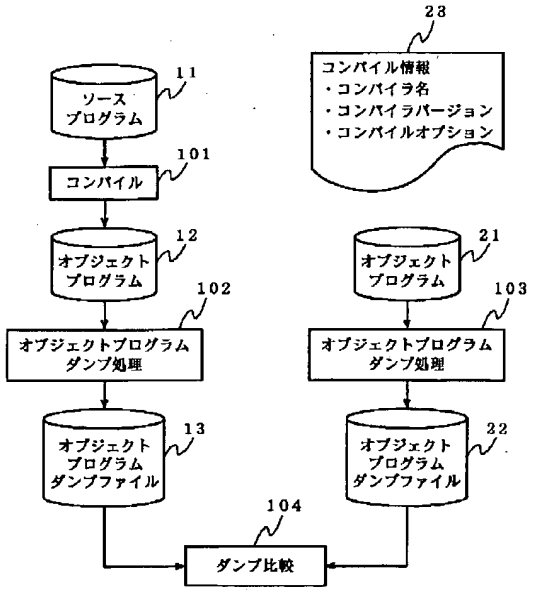
【図２】オブジェクトプログラムからコンパイル情報を抽出する処理手順を説明する図である。

【図３】ランユニットプログラムとオブジェクトプログラムとの関連付けを特定する処理手順を説明する図である。

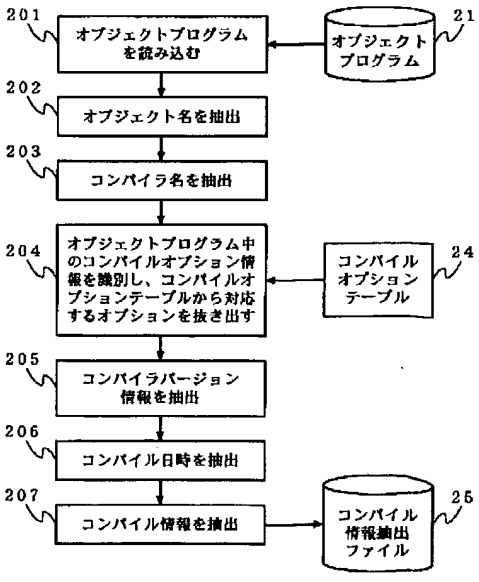
【符号の説明】

１１　ソースプログラム
１２，２１，３２　オブジェクトプログラム
１３，２２　オブジェクトプログラムダンプファイル
２３　コンパイル情報
２４　コンパイルオプションテーブル
２５　コンパイル情報抽出ファイル
３１　ランユニットプログラム
１０１～１０４，２０１～２０７，３０１～３０４　処理ステップ

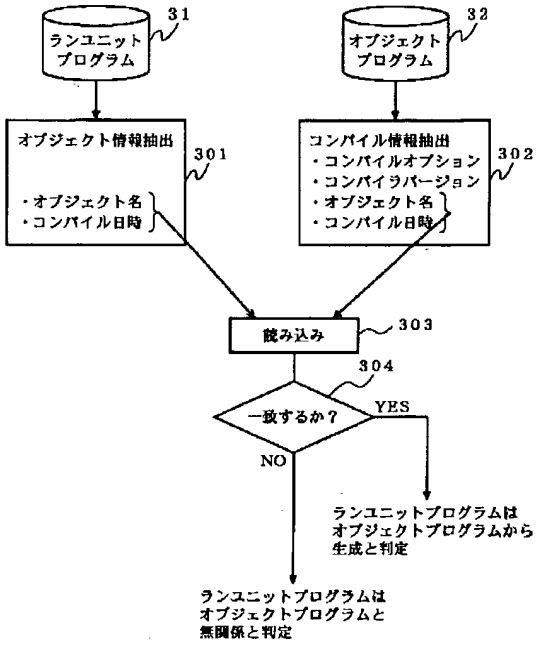
【図１】



【図２】



【図3】


ランユニット
プログラム
31
オブジェクト
プログラム
32
オブジェクト情報抽出
・オブジェクト名
・コンパイル日時
301
コンパイル情報抽出
・コンパイルオプション
・コンパイラバージョン
・オブジェクト名
・コンパイル日時
302
読み込み
303
304
一致するか？
YES
NO
ランユニットプログラムは
オブジェクトプログラムから
生成と判定
ランユニットプログラムは
オブジェクトプログラムと
無関係と判定